Introduction

The Controller adopts the technique of Utility Function Algorithm. The controlling process is considered under two separate scenarios:

1. When there are passengers inside the lift, their target floor is put into priority.
2. When there is no passenger inside the lift, the elevator will choose the next destination based on "Utility Function Algorithm"

Details of the Controller:

For scenario 1,

1. Identify the movement of the elevator, whether it is moving upwards, or downwards.
2. If the elevator is moving upwards, people whose target floor is higher than current floor will be considered firstly. And the policy is that the "closer the floor, the first to consider." If the elevator is moving downwards, the vice versa.

For scenario 2, I implemented the Utility Function.

1. Identify, outside the lift, on a floor, whether the "up request" button is pressed or "down request" button is pressed.
2. Running the Utility Algorithm to decide whether the lift should go upwards or downwards.
3. If the elevator chooses to go upwards, it will stop at floor where "up request" button is pressed firstly, the policy is that "the closer the floor, the first to go". If the elevator chooses to go downwards, the vice versa.

Quantitative Evaluation

When a passenger exits the lift, we deem that this passenger's goal is reached, as the passenger has been transported from its original position to its target position. Thus, when a passenger exits, the variable exit_person will increase by one. We evaluate the efficiency of the controller by counting how many people's goal has been reached within a period of time.

Result

In order to test the efficiency of the controller, I set the testing period to be 3000 ticks. Table 1 shows how many passengers' goal has been reached within 3000 ticks.

Table 1

	myController	Controller
Trail 1	160	160
Trail 2	157	152
Trail 3	164	151
Trail 4	154	153
Trail 5	154	156

Table 2

```
t = 1.3546, df = 7.811, p-value = 0.2134
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.412494  9.212494
sample estimates:
mean of x mean of y
    157.8     154.4
```

Table 2 shows the result of analysis of data in Table 1 by RStutio. It shows in terms of mean value, myController is performing slightly better. However, P-value has demonstrated that there is no significant difference between the two controllers.

One of the reason is that myController only improved scenario 1, where the lift could find an optimal destination based on the Utility Function Technique when there is no passenger in the lift. However, there is a very small probability when the lift is empty. To achieve the best performance, both passengers inside and outside should be considered at the same time. This, however, would increase the complexity of the implementing of the controller.

Future work is recommended to implement a controller which combines the target-floor of passengers inside and outside the elevator.

Code Instruction

Open “Mycontroller” document, running “simulator.py” file.   The Utility function if implemented in “mycontroller.py” file.